УДК: 004.8 (045) https://doi.org/10.15407/jai2023.03.139

В.М. Синєглазов[1], В.П. Хоцянівський[2]
[1,2]Національний авіаційний університет, Україна
вул. Любомира Гузара, 1, м. Київ, 03058
[1]svm@nau.edu.ua
[2]sttt912@yahoo.com
[1]https://orcid.org/0000-0002-3297-9060
[2]https://orcid.org/0000-0003-0415-777X

МЕТОД ПЛАНУВАННЯ РУХУ ПРОМИСЛОВИХ РОБОТІВ З ВИКОРИСТАННЯМ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ

Анотація. В роботі було розглянуто процеси планування та розгортання руху робота, шляхом розробки підходу до створення системи заснованої на нейронних мережах. Запропоновано систему яка вміє сприймати навколишнє середовище та керує рухом робота шляхом генерації коректних команд управління. Для цього було вирішено 3 задачі, а саме, аналіз середовища з метою визначення його ознак, визначення траєкторії з метою нейтралізації зіткнення, визначення керованих впливів для виконавчих органів з метою реалізації руху. Запропоновано функціонал та структуру нейронної мережі для вирішення кожного з завдань. Проведено порівняння запропонованого підходу з вже існуючими підходами по ключовим параметрах, таким як час виконання запланованого руху та час обчислення траєкторії руху.

Ключові слова: нейронні мережі, система планування руху, інтелектуальна система.

V. Sineglazov[1], V. Khotsyanivsky[2]
[1,2]National Aviation University, Ukraine
Liubomyra Huzara ave. 1, Kyiv, 03058
[1]svm@nau.edu.ua
[2]sttt912@yahoo.com
[1]https://orcid.org/0000-0002-3297-9060
[2]https://orcid.org/0000-0003-0415-777X

THE METHOD OF PLANNING THE MOVEMENT OF INDUSTRIAL WORK USING AN INTELLIGENT SYSTEM

Abstract. The paper considered the processes of planning and deployment of robot movement by developing an approach to creating a system based on neural networks. A system is proposed that can perceive the environment and controls the movement of the robot by generating correct control commands. For this purpose, 3 tasks were solved, namely, the analysis of the environment in order to determine its features, the determination of the trajectory in order to neutralize the collision, and the determination of controlled influences for the executive bodies in order to implement the movement. The functionality and structure of the neural network for solving each of the tasks is proposed. The proposed approach is compared with existing approaches on key parameters, such as the execution time of the planned movement and the time of calculating the movement trajectory.

Keywords: neural networks, motion planning system, intelligent system.

Вступ

Планування траєкторії руху робототехнічної системи є важливим викликом для галузей штучного інтелекту та робототехніки, і протягом багатьох років було запропоновано безліч методів для вирішення цього завдання з використанням методів оптимізації та евристичного пошуку. Однак, траєкторії, створені за допомогою цих підходів, зазвичай включають велику кількість проміжних точок і вимагають додаткової обробки для розгортання на промислових роботах.

У промисловій робототехніці траєкторії руху зазвичай програмуються за допомогою спеціалізованого Application Programming Interface(API), наданого виробником робота. Цей інструментарій визначає набір високорівневих команд руху, таких як «точка-точка», «лінійний» і «циклічний рух» [1]. Система керування роботом, розроблена виробником, має власний алгоритм пошуку шляху та контур

керування для виконання запрограмованого руху. Параметри керування точно налаштовані виробником робота. Зазвичай вони недоступні для користувача, але він може використовувати попередньо визначені команди руху та адаптувати їхні параметри для отримання бажаних рухів робота. Проте в деяких ситуаціях такий рух може призвести до зіткнення між роботом і динамічними перешкодами. Наприклад, у деякій системі програма планувальник керує кількома роботами, які будуть проходити через спільну зону в різні часові проміжки. Проте, через неправильно налаштовані параметри робот може зіткнутися з іншим роботом, якщо він увійде в спільну зону раніше або пізніше запланованого, наприклад через фізичну деформацію одного з своїх компонентів.

Тому виникає проблема планування руху робота користувачем на системах керування промисловими роботами з можливістю динамічної нейтралізації зіткнення. Тобто результати вказаного дослідження потрібні практиці, тому що вони визначають можливість безпечного та ефективного використання промислових роботів в умовах, де вони повинні взаємодіяти з динамічними оточуючими об'єктами, наприклад іншими роботами.

Постановка проблеми

Для робототехнічних систем із високими ступенями свободи простір розв'язків задачі планування руху є великим.

Беручи до уваги нелінійність середовища роботи робота, за основу системи пропонується обрати методи машинного навчання, оскільки вони дозволяють автоматизувати та спростити процеси калібрування рухомих елементів роботів, забезпечуючи максимальну точність та швидкість роботи. Наприклад, методи глибинного навчання можуть використовуватися для автоматичного визначення оптимальних параметрів руху робота, виходячи з даних, зібраних під час його роботи, а також для прогнозування можливих відхилень у його роботі [2].

Для цього необхідно створити нейронну мережу, яка повинна генерувати траєкторію руху робота без зіткнень з динамічними перешкодами у формі високорівневих команд руху [3], наприклад таких як переміщення на велику відстань. Ці команди руху можна легко перетворити на мову команд від виробника робота та імпортувати в систему керування роботом.

Нейронна мережа повинна використовувати реальну динаміку руху робота та траєкторію руху на основі фактичного виконання руху, щоб точно обчислити фактичні рухи робота та його траєкторію. Запланований рух робота може трохи відрізнятися від реального руху.

Все це дозволяє стверджувати, що доцільним є проведення дослідження, присвяченого розробці підходу до створення інтелектуальної системи планування руху промислових роботів.

Аналіз останніх досліджень і публікацій

В роботі [4] показано, що на більшості систем керування промисловими роботами береться за основу один із двох способів планування та розгортання руху роботів.

Найчастіше використовується спосіб, де користувач для деяких робототехнічних систем може використовувати лише попередньо визначені високорівневі команди руху та адаптувати їхні параметри для отримання бажаних рухів робота, наприклад рух від точки до точки. У цьому випадку більшість методів використовують випадкові позиції для зменшення кількості проміжних точок [5], або методи на основі оптимізації.

Прикладом використання таких способів є алгоритми інтерполяції, такі як лінійна інтерполяція. Якщо пряме з'єднання є безконфліктним, надлишкові проміжні точки можна усунути. Оскільки деякі параметри робота недоступні, наприклад динамічна поведінка та параметри керування, способи планування руху за допомогою алгоритмів зазвичай використовують завчасно пораховані

значення для розрахунку руху. Потім оброблену траєкторію перетворюють на попередньо визначені команди руху та імпортують в систему керування роботом.

Недоліком цього способу є те, що система керування роботом, яку надає виробник, виконує команди руху з використанням власного алгоритму пошуку траєкторії та параметрів керування. Ці параметри були спеціально розроблені та налаштовані для конкретного робота.

Але залишилися невирішеними питання, що ці параметри можуть відрізнятися від тих, що використовувалися в режимі моделювання. Це може призвести до геометричних і часових відхилень між запланованими та виконаними рухами. Геометричні відхилення можуть призвести до зіткнення робота з динамічними або нерухомими об'єктами.

Варіантом подолання відповідних труднощів може бути, використання системи керування роботами з додатковим інтерфейсом зв’язку, які дозволяють додатковому контуру керування керувати роботами використовуючи низькорівневі вхідні сигнали керування в режимі реального часу, такими сигналами можуть бути положення чи швидкість суглобів робота [6].

Саме тому, рухи роботів, що плануються двома описаними вище способами слід перевірити на етапі розгортання, щоб перевірити, чи геометричні та часові відхилення між плануванням і виконанням руху робота призводять до зіткнення. Коли відхилення призводить до зіткнення, фактичний рух робота потрібно скорегувати та перевірити знову.

Проте, більшість сучасних методів планування руху без зіткнень зосереджені на покращенні продуктивності алгоритмів планування руху в фазі планування. Через що, більшість сучасних досліджень зосереджені на вдосконаленні та перевірці продуктивності алгоритмів планування руху без зіткнень у середовищах моделювання, а не на тому, як розгорнути запланований рух робота в динамічному середовищі. Що є недоліком, оскільки сучасні гнучкі виробничі системи вимагають не лише автоматичного планування руху робота, але й можливості швидкого розгортання робота для його подальшої роботи.

Мета дослідження

Метою дослідження є розробка підходу до створення інтелектуальної системи планування траєкторії руху робота, яка забезпечить генерацію ефективних команд високого рівня з мінімальною кількістю проміжних точок, що можуть безпосередньо виконуватися системою керування.

Для досягнення даної мети були поставлені наступні задачі:

– створити підхід для генерації коректних траєкторій руху робота на різних відстанях;

– порівняти ефективність запропонованого підходу з алгоритмічними підходами, які використовуються в промислових системах.

Виклад основного матеріалу

У цій роботі запропоновано підхід зосереджений на вирішенні задач планування руху робота для підвищення ефективності пошуку траєкторії. Тобто, алгоритм повинен навчитись сприймати навколишнє середовище, для того щоб керувати рухом та передавати коректні команди управління.

Використання настільки багатовимірного представлення для налаштування системи є надзвичайно складним, тому необхідно цю складову розділити на декілька задач:

1. Задача аналізу середовища з метою визначення його ознак.
2. Задача визначення траєкторії з метою нейтралізації зіткнення.
3. Задача визначення керованих впливів для виконавчих органів з метою реалізації руху.

Розглянемо задачу аналізу середовища з метою визначення його ознак, для її вирішення доцільно використати згорткову нейронну мережу.

Така мережа може бути використана для розпізнавання, аналізу та інтерпретації великої кількості візуальної інформації, наприклад для аналізу динамічної та статичної сцени.

Динамічна сцена включає в себе рухомі об'єкти або зміну в стані об'єктів з часом. Нейронна мережа буде використовуватись для виявлення, відстеження, визначення швидкості та напрямку руху об'єктів [7].

Аналіз динамічної та статичної сцени з використанням нейронної мережі YOLOv7 є одним з ефективних підходів у галузі комп'ютерного зору. YOLO є однією з найпопулярніших архітектур для об'єктного виявлення та локалізації, і версія YOLOv7 є однією з останніх модифікацій цієї архітектури (рис. 1).

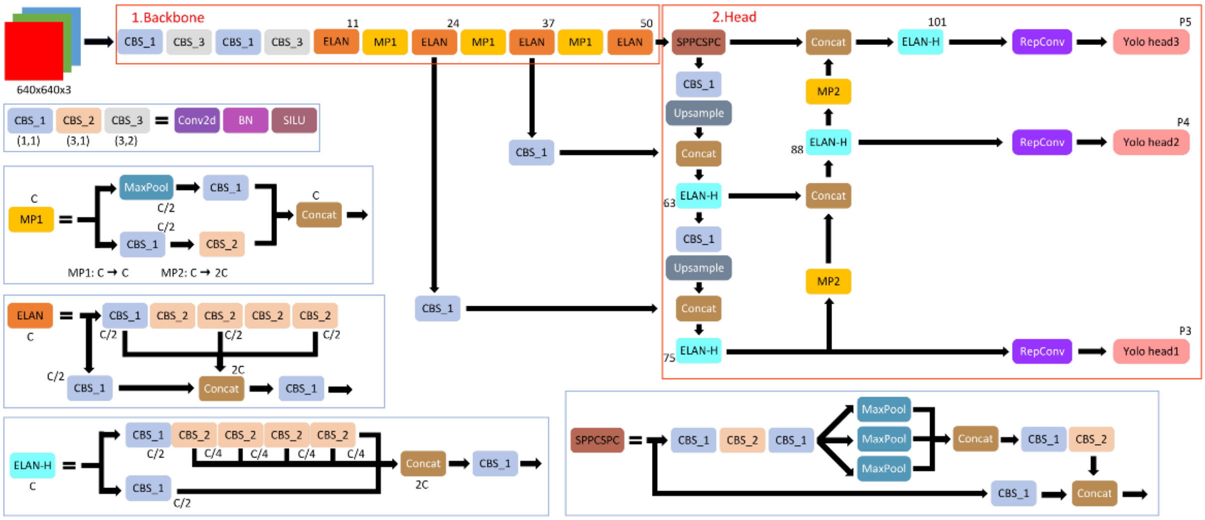


Рис. 1. Структура нейронної мережі YOLOv7

Для аналізу динамічної сцени з використанням YOLOv7, нейронна мережа може бути натренована на великій кількості відеозаписів з рухомими об'єктами. Після тренування, модель може використовуватись для автоматичного виявлення та відстеження рухомих об'єктів на відео.

Тому для вирішення задачі аналізу середовища з метою визначення його ознак, пропонується використати вказану архітектуру нейронної мережі, дані отриманні в результаті її роботи будуть передаватись в модулі, що вирішують наступні задачі.

Проте, аналіз середовища з метою визначення їх ознак є лише частиною завдання, після отримання ознак необхідно визначати траєкторію руху робота з метою нейтралізації зіткнення. Для цього пропонується також використати нейронну мережу, яка буде спрямована на забезпечення безпечної та ефективної роботи маніпулятора в навколишньому середовищі з обмеженими просторовими умовами. Така мережа буде базуватися на глибокому навчанні та використовується для автоматичного планування оптимальних траєкторій руху маніпулятора з метою уникнення зіткнень з перешкодами.

Опис функціональності нейронної мережі для визначення траєкторії з метою нейтралізації зіткнення:

1. Вхідні дані. Мережа приймає вхідні дані про поточне становище маніпулятора, такі як позиція та швидкість кожної з осей робота, а також інформацію про перешкоди в навколишньому середовищі.
2. Обробка даних. Вхідні дані проходять через шари нейронної мережі, де вони обробляються для виявлення закономірностей та взаємозв'язків між позиціями робота та перешкодами.
3. Траєкторія без зіткнень. Нейронна мережа виконує розрахунки та генерує оптимальну траєкторію руху маніпулятора, яка гарантує уникнення зіткнень з оточуючими перешкодами. Це досягається шляхом врахування геометричних

обмежень та меж безпеки для кожної з осей робота.

4. Вихідні дані. Після обробки вхідних даних та виконання розрахунків мережа надає вихідні дані, які використовуються для управління рухом маніпулятора. Ці дані включають значення позицій та швидкостей для кожної з осей, щоб забезпечити безпечний рух маніпулятора.

Переваги нейронної мережі для визначення траєкторії з метою нейтралізації зіткнення включають:

1. Здатність враховувати геометричні обмеження та межі безпеки робота.
2. Автоматичне планування оптимальних траєкторій з уникненням зіткнень.
3. Здатність адаптуватися до змінного середовища та різних сценаріїв руху.
4. Забезпечення безпечної та ефективної роботи маніпулятора у вузьких або складних умовах.

Застосування нейронної мережі визначення траєкторії робота маніпулятора дозволяє покращити безпеку та продуктивність робота, знизити ризик зіткнень та уникнути пошкоджень перешкод або самого робота.

Для вирішення завдання визначення траєкторії з метою нейтралізації зіткнень для робота маніпулятора пропонується використати рекурентну нейронну мережу (RNN). Основна ідея полягає в тому, що RNN (рис. 2) буде використовуватися для моделювання динаміки руху маніпулятора, здатної прогнозувати майбутні позиції осей робота на основі поточного стану, що виконується за допомогою Long Short-Term Memory (LSTM) шарів.

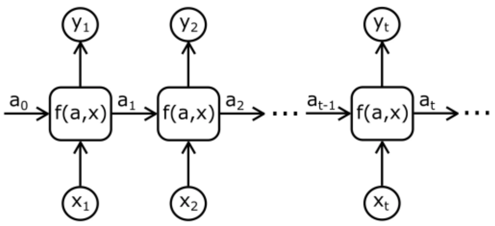


Рис. 2. Архітектура RNN

Структура RNN включає повторювані блоки, що дозволяють передавати інформацію через часові кроки. Один такий блок включає наступні компоненти:

1. Вхідний шар: Вхідні дані, такі як поточний стан робота, подаються на вхідний шар. У випадку визначення траєкторії це може бути вектор, що містить позиції та швидкості кожної осі маніпулятора.
2. Прихований шар: Прихований шар складається з нейронів, які виконують обчислення та передають інформацію в часових кроках. У випадку RNN це є LSTM (рис. 3). Цей шар дозволяє моделювати динаміку руху маніпулятора та враховувати залежності від попередніх станів.
3. Вихідний шар: Вихідний шар отримує інформацію з прихованого шару і генерує прогнозовані значення позицій кожної осі маніпулятора на майбутній часовий крок.
4. Зворотні зв'язки: RNN має зворотні зв'язки, що дозволяють передавати інформацію назад у часі. Це дозволяє моделі використовувати інформацію з попередніх часових кроків для кращого прогнозування.

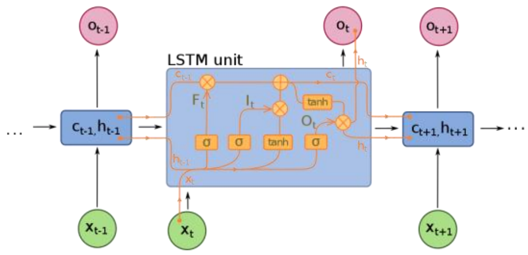


Рис. 3. Архітектура LTSM

Структура RNN для визначення траєкторії руху маніпулятора організована наступним чином:

1. Вхідний шар: Вектор, що містить поточний стан маніпулятора (позиції та швидкості кожної осі), подається на вхідний шар.
2. Повторюваний блок: Включає LSTM шар, який моделює динаміку руху маніпулятора та передає інформацію в часових кроках.
3. Вихідний шар: Генерує прогнозовані значення позицій кожної осі маніпулятора на майбутньому часовому кроці.

4. Зворотні зв'язки: Дозволяють передавати інформацію від вихідного шару назад до повторюваного блоку, щоб враховувати контекст з попередніх часових кроків.

5. Параметризація траєкторії: Вихідні значення позицій кожної осі можуть бути подані у вигляді параметрів траєкторії в якості цілочисельних значень.

Рухи робота, заплановані запропонованим підходом, мають загальний формат команд руху. Після генерації необхідної траєкторії необхідно визначити керуючий вплив та перетворити його в команди необхідні для реалізації руху робота або маніпулятора Тому, мережа визначення керованих впливів для виконавчих органів є важливою складовою системи керування рухом, оскільки вона вирішує завдання перетворення високорівневих команд або вказівок на конкретні керовані сигнали.

Структура мережі визначення керованих впливів наступна:

1. Вхідний шар: Вхідний шар отримує високорівневі команди або вказівки, що визначають бажаний тип руху або конкретні дії робота. Це такі величини, як напрямок, швидкість, прискорення або бажана позиція.

2. Приховані шари: Мережа включати приховані шари, які виконують обчислення та обробку інформації.

3. Вихідний шар: Вихідний шар мережі генерує керовані впливи, які необхідні для реалізації бажаного руху. Ці впливи подані у вигляді вектора, що керують відповідними виконавчими органами, а саме моторами.

4. Зворотні зв'язки: Мережа має зворотні зв'язки, які дозволяють адаптувати параметри мережі на основі отриманого вихідного сигналу та реальних результатів руху. Це покращило точність та забезпечити корекцію руху.

5. Параметризація впливів: Вихідні значення керованих впливів при потребі можуть бути параметризовані відповідно до вимог виконавчих органів, наприклад, як значення сил або керовані сигнали з певним діапазоном чи форматом.

Схематичну діаграму мережі визначення керованих впливів показана на рис. 4.

Порівняємо існуючий алгоритм планування руху робота, а саме Rapidly Exploring Random Trees (RRT) з шляхом який згенерувала спроєктована система (див. таб. 1).

Слід зазначити, що час виконання траєкторії руху робота істотно змінюється через різну відстань між початковою і кінцевою точкою руху.

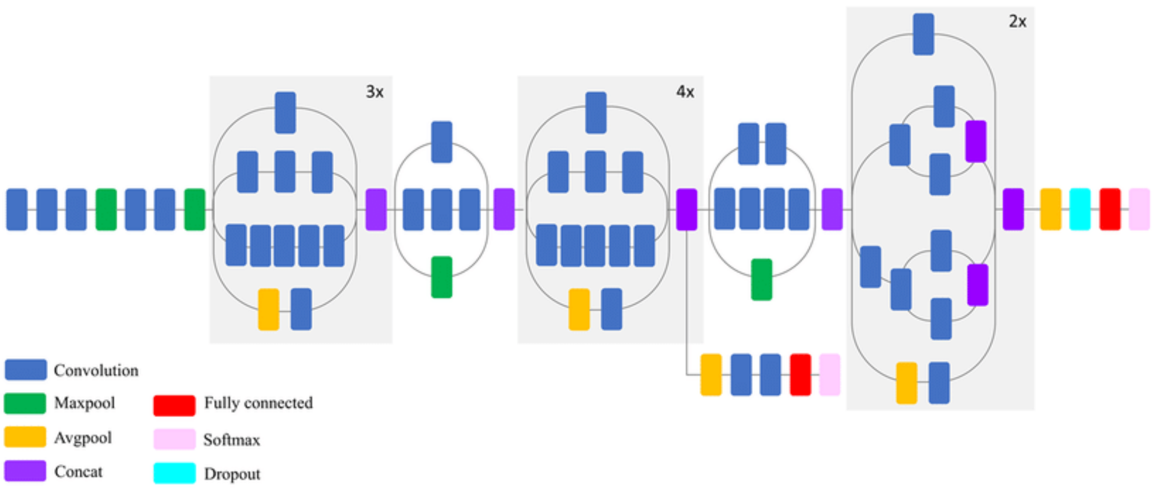


Рис. 4. Схематична діаграма мережі визначення керованих впливів

Таблиця 1. Порівняльна таблиця часу виконання траєкторій, згенерованих запропонованим підходом та RRT

Навколишнє середовище	Відстань між стартом і кінцем	Середній час виконання руху в мілісекундах	
		Запропонований підхід	RRT
Просте статичне середовище	Мала	221	212
	Середня	422	543
	Велика	659	836
Складне статичне середовище	Мала	291	372
	Середня	603	797
	Велика	732	904
Просте динамічне середовище	Мала	244	272
	Середня	496	581
	Велика	734	958
Складне динамічне середовище	Мала	419	462
	Середня	765	975
	Велика	1071	1294

Тому розділимо відстань між початковою і кінцевою точками руху в тестових прикладах на три класи:

1. Мала відстань (менше 30 % радіусу дії робота).
2. Середня відстань(більше 30 %, але менше 60 % радіусу дії робота).
3. Велика відстань (більше 60 % радіусу дії робота).

Також потрібно звернути увагу, що рух робота, запланований існуючою системою, значно відрізняється від руху робота, що запропонований спроєктованою системою керування. Це тому, що алгоритм керування RRT, який використовується на етапі планування, значно відрізняється від алгоритму керування описаного в статті, оскільки на етапі планування RRT припускає, що суглоби можуть досягти максимального прискорення. Однак насправді система керування роботом застосовує лише 60 % і 45 % максимального прискорення для осей робота [8]. Виходячи з цього, середній час виконання траєкторій, сформованих запропонованим підходом на двадцять відсотків швидший, ніж RRT.

Виходячи з цього, середній час виконання траєкторій, сформованих запропонованим підходом на двадцять відсотків швидший, ніж RRT.

Висновки

У результаті проведених досліджень була розроблена інтелектуальна система та підхід до планування траєкторії руху робота, що дозволяє генерувати ефективні команди високого рівня з мінімальною кількістю проміжних точок. Ця система базується на використанні нейронних мереж і може сприймати навколишнє середовище для коректного керування рухом робота та передачі відповідних команд управління.

Дослідження включало в себе три основні задачі: аналіз середовища для визначення його характеристик, планування траєкторії для уникнення зіткнень та визначення керованих впливів для виконавчих органів робота. Результати дослідження підтвердили високу ефективність запропонованого підходу до генерації траєкторій руху робота. З лише 5 недійсними траєкторіями серед 100 згенерованих середовищ, система демонструє високу точність у генерації траєкторії руху робота.

Література

1. Siciliano, B. and Sciavicco, L. (2016). Robotics: Modelling, Planning and Control. Springer. http://doi.org/10.1007/978-1-84628-642-1.
2. Orr, James & Dutta, Ayan. (2023). Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. https://doi.org/10.3390/s23073625
3. Rivlin, O. (2020, May 5). Generalized Planning With Deep Reinforcement Learning. arXiv.org. https://arxiv.org/abs/2005.02305
4. Niku, S.B. (2010). Industrial Robotics: Programming, Simulation and Applications. John Wiley & Sons. https://doi.org/10.5772/40
5. LaValle, S. M. (2006). Planning algorithms. Cambridge University Press. https://doi.org/10.1017/CBO9780511546877
6. Elhaki, O. and Shojaei, K. (2022). Output-feedback robust saturated actor–critic multilayer neural network controller for multi-body electrically driven

tractors with n-trailer guaranteeing prescribed output constraints. Robot. Aut. Syst. 154, 104106. https://doi.org/10.1016/j.robot.2022.104106

7. Jamshidi, Somayeh & Mirzaei, Mehdi & Malekzadeh, Maryam (2023). Applied nonlinear control of spacecraft simulator with constraints on torque and momentum of reaction wheels. ISA Transactions. http://doi.org/10.1016/j.isatra.2023.03.027

8. Wong, C.C., Chen, C.J., Wong, K.Y., Feng, H.M. (2023). Implementation of a Real-Time Object Pick-and-Place System Based on a Changing Strategy for Rapidly-Exploring Random Tree. Sensors, 23(10), 4814. https://doi.org/10.3390/s23104814

References

1. Siciliano, B. and Sciavicco, L. (2016). Robotics: Modelling, Planning and Control. Springer. http://doi.org/10.1007/978-1-84628-642-1

2. Orr, James & Dutta, Ayan. (2023). Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. https://doi.org/10.3390/s23073625

3. Rivlin, O. (2020, May 5). Generalized Planning With Deep Reinforcement Learning. arXiv.org. https://arxiv.org/abs/2005.02305

4. Niku, S.B. (2010). Industrial Robotics: Programming, Simulation and Applications. John Wiley & Sons. https://doi.org/10.5772/40

5. LaValle, S. M. (2006). Planning algorithms. Cambridge University Press. https://doi.org/10.1017/CBO9780511546877

6. Elhaki, O. and Shojaei, K. (2022). Output-feedback robust saturated actor–critic multilayer neural network controller for multi-body electrically driven tractors with n-trailer guaranteeing prescribed output constraints. Robot. Aut. Syst. 154, 104106. https://doi.org/10.1016/j.robot.2022.104106

7. Jamshidi, Somayeh, Mirzaei, Mehdi & Malekzadeh, Maryam. (2023). Applied nonlinear control of spacecraft simulator with constraints on torque and momentum of reaction wheels. ISA Transactions. http://doi.org/10.1016/j.isatra.2023.03.027

8. Wong, C.C., Chen, C.J., Wong, K.Y., Feng, H.M. (2023). Implementation of a Real-Time Object Pick-and-Place System Based on a Changing Strategy for Rapidly-Exploring Random Tree. Sensors, 23(10), 4814. https://doi.org/10.3390/s23104814

The article has been sent to the editors 07.10.23.
After processing 20.10.23.
Submitted for printing 30.11.23.